Scenario-Based Design Methods for the Development of Pen-Based Software User Interface*

Danli Wang, Guozhong Dai, Hongan Wang
Institute of Software
Chinese Academy of Sciences
Beijing, China

Steve C. Chiu
College of Engineering
Idaho State University
Pocatello, ID, U.S.A.

Abstract

Pen-based user interface (PUI) has drawn significant interest, owing to its intuitiveness and convenience. While much of the research focuses on the technology, the usability of a PUI has been relatively low since human factors have not been considered sufficiently. Scenario-centric designs are ideal ways to improve usability. However, such designs possess some problems in practical use. To cope with these design issues, the concept of "scenarios" is proposed in to facilitate the interface design, and to help users understand the interaction process in such designs. The proposed scenario-based approach for PUI design is coupled with a practical application to show its effectiveness and usability.

Keywords: HCI, Scenario-Based Design, UI.

1 Introduction

With the rapid advancement in information and network technology, computers are becoming ubiquitous, and the role of Human-Computer Interaction (HCI) is becoming increasingly important. As the link between human and computer, interface is one of the most important components of an interactive system. The main objective of HCI research is to develop natural, harmonic and efficient user interfaces, so as to realize human-centered interaction [1], and make "computers adapted to human". The center of this interface design is changing from the paradigm of "easy realization" to the new "easy learning and high usability".

Pen-based User Interface (PUI) is an effective method to make the interface natural. The first PUI appeared in the Sketchpad in 1963 [14]. Since the middle of 1980s, PUI-based products such as PAD developed rapidly, accompanying the computing revolution, including distributed computing and mobile computing. There are a lot of research achievements on PUI at MIT, CMU, Washington University, etc. At the same time, Microsoft, Wacom, Toshiba and NEC moved into the PUI fields quickly [9]. In China, many research institutes and universities are also carrying out the research on PUI, and they have gained achievements in many aspects, such as word processing, PUI framework, and multi-modal interaction, to name a few.

PUI provides users a natural interaction method, helping people capture their thoughts conveniently and think visually. Compared with the traditional WINP method (i.e. Window, Icon, Menu, Pointer), PUI changes input information type from discrete, explicit digital data to continuous, implicit analog data; and it changes the input methods from keyboard and mouse to pen. In view of the difficulties of inputting Chinese through keyboard caused by the hieroglyphic feature of Chinese, PUI, as a promising way to solve this problem, presents significant promise in China.

Owing to its naturalness and convenience, PUI has attracted much attention, and is being employed in more and more applications [9]. However, much of the current research focuses on the implement technology, failing to give sufficient consideration to human factors, which, together with the problems in development methodology, makes a negative impact on development efficiency and usability. Therefore, how to develop user-content PUI software deserves more concern.

Scenario-based design (SBD) represents a desirable method to enhance software usability. The focus of SBD centers on the description about the users, and how the users perform the tasks [4]. It is a good way to extract users' demands; and provides a tool to design products with high usability. However, SBD have its own disadvantages.

To address these problems, the concept of "interface scenarios" is proposed in this paper to facilitate interface design, and to help users understand the mutual interaction process. A scenario-centric methodology for PUI development is also proposed along with a practical application to show its effectiveness.

* This research is supported in part by the National Grand Fundamental Research 973 Program (under Grant No. 2002CB312103), and the National Natural Science Foundation of China (under Grant No. 60373056).

2 Related Work

2.1 Scenario-Based Design

Scenarios are some stories about people and their activities [3]. Let's see a concrete example: A man wants to find a meeting agenda of June 2, 2005. *First he turns on his computer and enters document management application. Then he opens his routine file folder, in which he finds the meeting agenda folder. Finally, he locates and opens the agenda file in the agenda folder.* Scenarios highlight the purpose and the usage of the system through detailed description of the human-machine interaction process: what the system is used for, what processes are involved, whether these processes are carried out successfully, and how people will react to them.

Scenario-based design is a design methodology that considers scenarios as a central artifact in system design. The approach encourages user involvement in system design, provides shared vocabulary among the people participating in the system development project, envisions the uncertain future tasks of the end users, and enhances ease of developing instructional materials. It provides a good brainstorming tool for planning and helps all the stakeholders to choose options in their decision-making [6].

Scenario-based design, which includes user participation in the design process, is believed to increase product success through valuable user insight. Based on these advantages of scenario-based design, scenarios form the foundation for the design, development, and evaluation of a system.

The earliest successful application of scenario-based design was the voice message system developed by IBM for the L.A. Olympic Games [7]. Since then, the design found its way in various types of software development. In HCI, scenarios are used to describe detailed context to facilitate the design decision-making [4]. In software engineering, use cases are employed to depict the situations of a system in use, while scenarios are the instances of use cases [8]. Requirement engineering uses scenarios to record the observance and analysis by the users, from which the requirements are extracted. Scenarios can also be used in requirement assessment [12].

The application of scenarios-based design methods has been rather successful in HCI. Prof. Carroll's team did some excellent work with it. Rosson *et al*. proposed a scenarios-based framework for interaction systems [13], which integrated scenarios analysis and design in the phases: analysis, design, prototype and assessment. In the analysis phase, problem scenarios describe the practical problems; in the design phase, action scenarios, information scenarios and interaction scenarios depict the problems in design; in the prototype and assessment phase, scenarios are used for assessing the usability of the prototype. This method enhanced the usability of interaction systems.

Although scenario-based design is an excellent interactive method, a scenario itself does not provide a good way to implement user-centered design, while how to obtain user requirements from scenarios is the pivot of the design. Among the existing applications of scenarios-based design, though some of them mentioned user-centered designs, as a whole they still consider user requirements from the system's point of view, falling short of really designing systems from the users' perspective.

2.2 Pen-Based User Interface

Broadly speaking, the research and development of PUI went though three major phases: 1. the metamorphosis of WIMP; 2, establishing the interface model on the basis of sketching; 3, wide application. The first phase of PUI development occurred between the end of 1980s and the beginning of 1990s, during which PUI applications were mainly seen in mobile digital computing devices with pen superseding mouse in clicking function.

The second phase was from the beginning of 1990s to the end of 1990s. The milestone was Xerox PARC's LiveBoard [5], a white board sized pen-based interaction device, and its related software, which showed many fundamental ideas and concepts of modern PUI research. PUI research then opened a new page. The research focus during this phase was user-centric design, naturalness and efficiency. It gave birth to the brand new sketching-based PUI interaction model. Besides the LiveBoard, other creative and usable systems developed during this period include CMU's SILK [10], which is used in GUI prototype design, and Flatland [11] by Georgia Tech.

The third phase spanned from the end of 1990s till now. With the advancement of both hardware and software, PUI reached its mature stage and is expanding into wider application fields: pen-based sketching tool for animation design, children intelligence exploiting entertainment software, software for clinical devices and education software, to name just a few. Microsoft went even further as to develop an operating system with full handwriting input support for their Tablet PC, which made the debut on the market at the end of 2002. In recent years, Microsoft Asia R&D developed the so-called digital ink technology, which made handwriting easier to read, access, organize and use [15]. In addition, Swedish Anoto AB developed blue tooth-based digital pens and digital papers, which presented a promising prospect for PUI R&D and future applications [2]. In short, PUI has seen tremendous growth since its origination in various end-use areas.

3 Interface Scenario

Scenarios can be depicted by various methods: text, graph, story board, prototype, etc. [13] After analyzing these methods, in view of the problems in PUI design, we propose the concept of "interface scenario". This kind of scenario uses the designed interface as the background, and depicts the operations needed for a task explicitly on the background by texts and/or graphs, giving users an intuitive grasp on the interface operation and facilitating interface evaluation. Figure 1 shows a simple example:

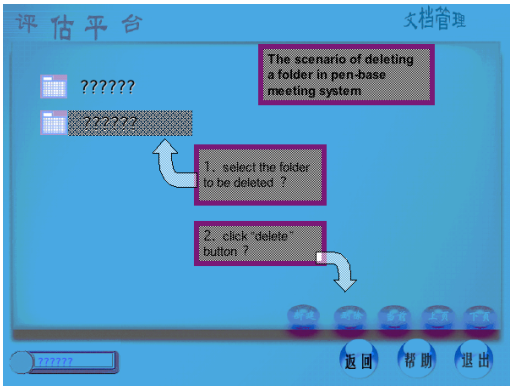

Figure 1. An interface scenario

This scenario describes a user deleting a document in an evaluation system. This scenario not only helps the user understand the system easily by giving them an intuitive description of what the system will look like and how it will behave, but also facilitates the communication between the designers and the users.

4 Scenario-Based Development Method for Pen-Based User Interface

Based on the characteristics of PUI and scenario-based design, we present our method for the PUI software development, as shown in Fig. 2.

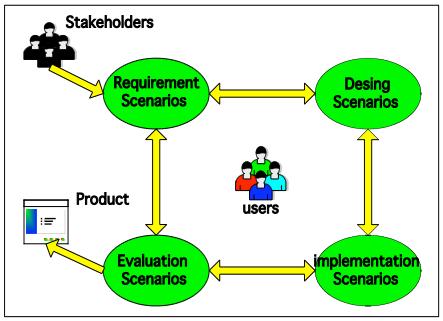

Figure 2. PUI development process

4.1 Development Procedure

We specify the following procedure for scenario-based PUI development:

Requirements: Using these scenarios, we described the interviews and observations. Based on the observation scenarios, we established the requirements model and evaluated it through discussions with the users and experts.

Design: Analyzing the results of requirements analysis, we designed the interface and the interface scenarios. Then we give the design solution.

Implementation: Based on the design solution, we coded and implemented a prototype.

Evaluation: Analyzing the system from the point of view of usability, we designed the scenarios for evaluation, and evaluated the prototype. Following this evaluation process, we modified the design when necessary

5 Example – A Pen-Based Meeting System

Communicating and discussing is a very good way to synthesize group intelligence, fostering creative collaborations. The traditional group discussion basically is first to elucidate content using, e.g. PowerPoint in meeting room, and then to discuss the problems with the help of a whiteboard/blackboard. PowerPoint is quite powerful and very nice for general use, enjoying an advantage especially in formal meetings. But in informal discussions, it shows some drawbacks: it is rather difficult to modify the document in an ad hoc manner during the discussion, such as recording the new ideas. It is especially hard to those who are unfamiliar with keyboards. To address this issue, we designed a pen-based meeting system, and developed a prototype.

5.1 Requirements Analysis

Through investigating the related users and analyzing the existing meeting systems, we set up the meeting system's scenarios description and functional model. Figure 3 shows a scenario of preparing a report for an HCI meeting.

Meeting report preparation

1. Click and enter the root interface of the meeting system
2. Enter the meeting management interface and find the HCI meeting
3. Click and enter the HCI meeting interface
4. Click report preparation icon to enter its interface
5. Edit the report in that interface

Figure 3. Meeting reporting scenario

The meeting system here has three major functions: meeting preparation (including meeting schedule and report preparation), meeting in process (including meeting registration, report playing, and report modifications), and post-meeting compilation.

5.2 User Interface Design

We then designed the interfaces and corresponding interface scenarios. Figures 4–7 illustrate several example interfaces and interface scenarios.

Figure 4. Meeting management UI

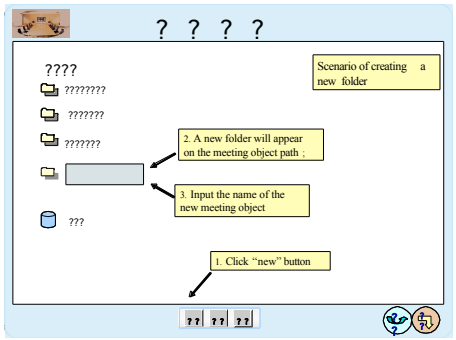

Figure 5. New folder creation

Figure 6. **UI of meeting playing**

Figure 7. Scenario of meeting playing

5.3 Evaluation of Design

We evaluated both the old and the new meeting systems. Twelve experts attended the evaluation. We obtained the evaluation results, which are shown in Table 1, from the statistic analysis of the data collected from the 10 expert assessment questionnaires we received using Excel and SPSS 10.0, and the analysis of the interview records.

Table 1: Average score on each evaluation dimension (min: 1, max: 6)

Evaluation Dimension		Average
New system vs. old system	Satisfaction	4.40
	Learnability	4.60
	Usability	4.10
	Interface	4.40
	Convenience in switching	4.40
	Learnability in switching	4.40

We made further statistic analysis using the unique sample T-test statistical method to find out on which dimensions the software received positive assessment. The value results are shown in Table 2.

Table 2 Analysis of the significant level

Evaluation dimension		T value	Degree of freedom	Significant level, p
New system vs. old system	Satisfaction	2.946	9	.016
	Learnability	3.601	9	.006
	Usability	1.908	9	.089
	Interface	5.511	9	.000
	Convenience in switching	5.511	9	.000
	Learnability in switching	3.375	9	.008

This table shows that, compared with the old system, the new system received positive assessment on the overall level. On the dimensions of learnability and interface, significant level was reached. Though the usability of the new system is assessed positively, it failed to reach the significant level, which can be attributed to the usability of the new system. The software is still in the process of being improved.

6 Conclusion

PUI software draws interest due to its naturalness and convenience. To enhance the development efficiency and usability of PUI software, we proposed the concept of interface scenario, and a development method for PUI software. A practical example proved the methodology's effectiveness, showing that both the development time and development cost to be reduced, while the usability has been enhanced.

Acknowledgement

The authors appreciate the evaluation performed by Liu Rude, Chai Songzhen, Niu Wenjia, Xu Juan, Cheng Tiegang, and the implementation of the meeting system by Feng Tian, Fei Su, Yanyan Qin and Lishuang Xu.

References

1. Abowd G. D. and Atkeson C. G. et al., "Teaching and learning as multimedia authoring: the classroom 2000 project," Proceedings of the fourth ACM international conference on Multimedia, Boston, USA, 1996:187-198.
2. Anoto AB Company [OL], http://www.anoto.com, 2003.
3. Carroll, J. M., "Five Reason for Scenario-based Design," Proceedings of the 32nd Hawaii International Conference on System Sciences, 1999.
4. Carroll J. M., "Scenarios and Design Cognition," APCHI2002, Oct.23-26, Beijing, China, 2002: 23-46.
5. Elrod S. and Bruce R et al., "LiveBoard: A Large Interactive Display Supporting Group Meetings, Presentations, and Remote Collaboration," Proceedings of the ACM Conference on Human Factors in Computing Systems: CHI'92, Monterey, Canada, 1992:599-607.
6. Go K., Carroll J. M. and Imamiya A., "The Blind Men and the Elephant: Views of Scenario-Based System Design," J. Interactions, 2004: Vol. 6, pp. 45-53.
7. Gould J. D., "How to design usable Systems," Readings in Human-Computer Interaction: Toward the Year 1995: 93-121.
8. Jacobson I., Christersson M., Jonsson P., and Overgaard, G., "Object-Oriented Software Engineering: A Use-Case Driven Approach." Addison-Wesley, Reading, MA, 1992.
9. Li Y., "Research on Pen-based User Interfaces—Theory, Technique and Implementation, "PhD dissertation, 2002
10. Landay, J. A. and Myers, B. A., "Sketching Interfaces: Toward More Human Interface Design." IEEE Computer, Vol. 34, No.3, pp. 56-64, March 2001.
11. Mynatt, E. D., Igarashi, T., Edwards, W. K. and LaMarca, A. Flatland: New Dimensions in Office Whiteboards." In Proceedings of CHI'99 Human Factors in Computing Systems, ACM, 1999: 346-353.
12. Potts C., Takahashi K. and. Anton A. I., "Inquiry-based Requirements Analysis." IEEE Software, 1994, 11(2): 21-32.
13. Rosson, M. B. and Carroll, J. M., "Scenario-Based Design." In: Jacko J. A. and Sears A., The Human-Computer Interaction Handbook. New Jersey: Lawrence Erlbaum Associates, 2002: 1032-1050.
14. Sutherland, I.E. "SketchPad: A Man-Machine Graphical Communication System," in AFIPS Spring Joint Computer Conference. 1963: 329-346.
15 Wang J., "Human-computer interaction research and practice in China" J. Interactions, 2003, 10(2): 88-96